

ANALISIS ULASAN SENTIMEN APLIKASI *MOBILE JKN* DENGAN ALGORITMA *SUPPORT VECTOR MACHINE* BERBASIS *PARTICLE SWARM OPTIMIZATION*

Nida Maulida, Nana Suarna, Willy Prihartono

Teknik Informatika, STMIK IKMI Cirebon
Jalan Perjuangan No. 10B Majasem Kota Cirebon
maulidanida4@gmail.com

ABSTRAK

Inovasi elektronik dalam layanan jaminan kesehatan pemerintah dikenal sebagai aplikasi *Mobile JKN* yang memudahkan peserta Jaminan Kesehatan Nasional-Kartu Indonesia Sehat (JKN-KIS) untuk mendapatkan layanan dan informasi. Dengan inovasi ini, ada banyak pro dan kontra sehingga banyak komentar muncul di kolom *review Google Play Store*. Kecenderungan respon pengguna dalam menggunakan aplikasi *Mobile JKN* dapat diketahui dengan analisis sentimen. Analisis sentimen adalah sistem untuk mengenali dan mengekstraksi *review* dalam bentuk teks. Penelitian ini bertujuan untuk mengukur tingkat akurasi, presisi, *recall* dan *Area Under Curve* (AUC) yang dihasilkan dari algoritma *Support Vector Machine* (SVM) berbasis *Particle Swarm Optimization* (PSO) terhadap aplikasi *Mobile JKN* di *Google Play Store*. Penelitian dilakukan dengan menggunakan algoritma *Support Vector Machine* (SVM) yang dioptimasi menggunakan *Particle Swarm Optimization* (PSO). Penelitian ini melibatkan berbagai tahapan *preprocessing*, seperti *cleaning*, *case folding*, *tokenizing*, dan *filtering* (*stop-word removal*). Hasil penelitian yang menggunakan 10 *k-fold Cross Validation* pada algoritma *Support Vector Machine* (SVM) yang menggunakan seleksi fitur *Particle Swarm Optimization* (PSO) menunjukkan hasil akurasi, presisi, *recall* dan *Area Under Curve* (AUC) secara berurutan sebesar 89,53%, 88,17%, 45,96% dan 0,869 dengan kategori (*Good Classification*).

Kata kunci : Aplikasi *Mobile JKN*, Analisis Sentimen, *Support Vector Machine* (SVM), *Particle Swarm Optimization* (PSO)

1. PENDAHULUAN

Semua orang di Indonesia memiliki program Jaminan Kesehatan Nasional yang dikelola oleh Badan Penyelenggara Jaminan Sosial (BPJS). BPJS Kesehatan melalui Program Jaminan Kesehatan Indonesia Sehat (JKN-KIS) berupaya memperluas cakupan dengan target mencakup seluruh masyarakat Indonesia sehingga mencapai *Universal Health Coverage* (UHC) [1]. Menurut Ali Ghufroon Mukti, Direktur Utama BPJS Kesehatan, jumlah peserta Jaminan Kesehatan Nasional-Kartu Indonesia Sehat (JKN-KIS) mencapai 252,17 juta orang pada Maret 2023, yang merupakan 90,79 persen dari total populasi Indonesia.

Dengan meningkatnya jumlah peserta, peserta JKN-KIS sering menghadapi masalah seperti antrian panjang untuk pendaftaran dan layanan kesehatan lainnya. Di era globalisasi ini, mendorong BPJS Kesehatan untuk mengembangkan aplikasi *mobile* bernama *Mobile JKN* untuk memudahkan pengguna dalam registrasi dan pengelolaan kesehatan. Aplikasi ini telah diperkenalkan sejak tahun 2017 [2] dan dapat diunduh di *Play Store*, *Google Play Store* dan *App Store* selaku penyedia layanan aplikasi telah menyediakan kolom *review* sehingga pengguna dapat memberikan pendapat terhadap aplikasi [3]. Namun, pengguna saat ini menghadapi beberapa masalah dengan *Mobile JKN* seperti: aplikasi sering mengalami masalah saat digunakan, tidak dapat digunakan jika tidak diupdate, dan sering meminta untuk diupdate versi terbaru bahkan setelah diupdate, menyebabkan pengguna tidak dapat *login* [4].

Berdasarkan studi literatur yang peneliti amati, penelitian yang dilakukan oleh Primandani Arsi, Rizki Wahyudi, Retno Waluyo. Penggunaan algoritma pemilihan fitur PSO pada klasifikasi model SVM disarankan untuk meningkatkan akurasi analisis sentimen terkait perpindahan ibu kota. Dengan menggunakan data dari 1.319 *tweets* dengan 457 sentimen positif dan 862 sentimen *negative* penelitian menemukan bahwa akurasi kategori "*fair classification*" meningkat sebesar 81,15% dari 79,06% sebelumnya [5].

Penelitian yang dilakukan Noor Hafidz dan Dewi Yanti Liliana membahas tentang klasifikasi sentimen COVID-19 di *Twitter*. Kombinasi algoritma *Support Vector Machine* (SVM), *N-Gram*, dan *Particle Swarm Optimization* (PSO) digunakan untuk menilai kinerja model klasifikasi dengan menggunakan nilai akurasi 0,755, presisi 0,719, *recall* 0,837, dan *area under the ROC Curve* (AUC). Eksperimen yang dilakukan menunjukkan bahwa kombinasi SVM, *N-Gram*, dan PSO sangat baik dalam mengklasifikasikan sentimen tweet dengan nilai akurasi sebesar 0,755, presisi sebesar 0,719, dan *recall* sebesar 0,837 [6].

Penelitian selanjutnya yang dilakukan oleh Anas Faisal, Yuris Alkhalifi, Achmad Rifai, dan Windu Gata memanfaatkan PSO untuk mengoptimalkan algoritma SVM dan *Naive Bayes*. Tanggapan anggota DPR RI di media sosial *Twitter* terkait dengan gagasan kebijakan atau bersifat terbatas. Berdasarkan pengujian validasi *k-fold*, PSO mampu meningkatkan akurasi SVM tertinggi berdasarkan PSO sebesar 3,99% dan akurasi *Naive Bayes* sebesar 2,8% [7].

Berdasarkan uraian di atas, maka dalam penelitian ini algoritma *Support Vector Machine* (SVM) akan digunakan dalam penelitian. Optimasi *Particle Swarm Optimization* (PSO) juga digunakan untuk mencoba meningkatkan nilai akurasi dari algoritma yang digunakan. Penelitian ini akan mengevaluasi kualitas hasil analisis algoritma baik dengan menggunakan optimasi *Particle Swarm Optimization* (PSO) maupun tanpanya dengan membandingkan nilai akurasi, presisi, *recall* dan AUC.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian pertama membahas tentang perbandingan metode *Naive Bayes* dan SVM untuk analisis sentimen terhadap vaksin *Astrazeneca* di *Twitter*. Tahapan penelitian ini meliputi *scraping data*, *preprocessing*, ekstraksi fitur, pelabelan, pembuatan model klasifikasi, dan evaluasi performa klasifikasi. Hasil penelitian menunjukkan bahwa metode SVM lebih akurat sebagai metode pengelompokan dalam proses analisis sentimen masyarakat tentang vaksin *Astrazeneca* di *Twitter* daripada metode *Naive Bayes*. Hasil penelitian yang menggunakan *Support Vector Machine* menghasilkan akurasi sebesar 87,27% dari data uji dan akurasi sebesar 76,81% untuk metode *Naive Bayes* [8].

Penelitian kedua membahas sentimen analisis pada *review* pengguna *myIndiHome*. Penelitian ini menggunakan algoritma SVM dan *Naive Bayes Classifier*. Data ulasan sebanyak 2.539 dikumpulkan dari 1 November 2020 hingga 15 Desember 2020, dengan 1.160 ulasan dalam kelas sentimen negatif dan 1.374 ulasan dalam kelas sentimen positif. Hasil menunjukkan bahwa kesalahan layanan *IndiHome* masih cukup tinggi, mencapai 46,7%, dengan banyak ulasan negatif. Hasil klasifikasi menunjukkan bahwa metode *Support Vector Machine* (SVM) memiliki nilai akurasi total rata-rata 86,54% lebih besar daripada metode *Naive Bayes Classifier* (NBC), yang memiliki nilai akurasi total rata-rata 84,69% [9].

Penelitian ketiga membahas persepsi masyarakat terhadap jenis vaksin COVID-19 yang diberikan di Indonesia. Penelitian ini menggunakan teknik analisis sentimen dengan menggunakan algoritma *Naive Bayes* (NB), *Support Vector Machine* (SVM), dan *Long Short-Term Memory* (LSTM). Pra-pemrosesan melibatkan *casefolding*, *tokenizing*, *filtering*, *stemming*, dan pembobotan kata menggunakan TF-IDF sebelum pemodelan. Setelah itu, pengujian model dilakukan dengan *Cross Validation* menggunakan *Python*, dan hasilnya dievaluasi dan divalidasi menggunakan *Confusion Matrix*. Hasil penelitian menunjukkan bahwa metode SVM memiliki skor akurasi 84,89% untuk model terbaik, sedangkan algoritma *Naive Bayes* memiliki skor akurasi 84,65% dan algoritma LSTM memiliki skor akurasi 82,97% [10].

2.2. Analisis Sentimen

Analisis sentimen yang juga dikenal sebagai *opini mining* adalah kombinasi *data mining* dan *text mining* yang digunakan untuk mengevaluasi pendapat, sentimen, evaluasi, sikap, penilaian, perasaan, dan emosi seseorang untuk memastikan apakah pembicara atau penulis setuju dengan topik, produk, layanan, organisasi, individu, atau kegiatan tertentu. Sistem yang bertujuan untuk menemukan dan mengekstrak pendapat dalam teks dibangun melalui analisis sentimen. Hal ini menunjukkan hubungan antara penambahan teks dan klasifikasi teks. Polaritas klasifikasi mengacu pada bentuk paling dasar, yaitu apakah sebuah kalimat atau teks berisi informasi negatif atau positif [7].

2.3. Mobile JKN

Aplikasi *Mobile JKN* merupakan suatu platform layanan kesehatan digital yang dikembangkan oleh BPJS (Badan Penyelenggara Jaminan Sosial) Kesehatan. Tujuannya adalah untuk membuat pengguna Jaminan Kesehatan Nasional-Kartu Indonesia Sehat (JKN-KIS) memiliki akses yang lebih mudah. Inovasi teknologi ini diharapkan dapat diterima oleh masyarakat. Penting untuk melakukan jajak pendapat masyarakat untuk mengetahui reaksi masyarakat terhadap layanan BPJS pasca peluncuran aplikasi *Mobile JKN*.

Aplikasi *Mobile JKN* meliputi informasi kepesertaan, perubahan data pelanggan, pembayaran termasuk kartu registrasi pelanggan, manfaat, informasi metode pembayaran donasi, catatan pembayaran termasuk riwayat transaksi pembayaran, virtual account, riwayat layanan dan layanan pendaftaran, verifikasi, informasi JKN, lokasi dan pengaduan. Banyak fitur aplikasi ini membantu peserta BPJS mendapatkan layanan kesehatan di tingkat pertama dan tingkat lanjut [1].

2.4. Term Frequency-Inverse Document Frequency (TF-IDF)

Istilah pembobotan digunakan untuk meningkatkan analisis sentimen dalam pemrosesan teks. Dalam penelitian ini, *Term Frequency-Inverse Document Frequency* (TF-IDF) digunakan dalam *RapidMiner* dan dilakukan dengan menggunakan operator *Process Document From Data*. Frekuensi suatu kata ($tf(w,d)$) diasumsikan bergantung pada seberapa sering kata tersebut muncul dalam teks atau dokumen. Untuk memantau kemunculan token dalam kalimat teks, gunakan metode pembobotan token frekuensi dokumen terbalik (IDF).

Persamaan (1), (2), dan (3) dapat digunakan untuk menghitung nilai TF-IDF.

$$Wtf_{t,d} = \begin{cases} 1 + \log_{10} tf_{t,d}, & \text{if } tf_{t,d} > 0 \\ 0, & \text{if } tf_{t,d} = 0 \end{cases} \quad (1)$$

$$idf_t = \log_{10} \frac{N}{DF_t} \quad (2)$$

$$W_{t,d} = Wtf_{t,d} \cdot idf_t \quad (3)$$

Keterangan:

- $W_{tf_{t,d}}$ = Bobot kata di setiap dokumen
- $tf_{t,d}$ = Jumlah kemunculan term dalam dokumen
- N = Jumlah keseluruhan dokumen
- DF = Jumlah dokumen yang mengandung term
- idf = Bobot inverse dalam nilai df
- $W_{t,d}$ = Pembobotan TF-IDF

2.5. Knowledge Discovery in Database (KDD)

Knowledge Discovery in Database (KDD) merupakan metode pengambilan pengetahuan dari database yang sudah ada yang terdiri dari tabel-tabel yang saling berhubungan/berkorelasi. Proses ini menciptakan pengetahuan yang dapat digunakan sebagai basis pengetahuan untuk tujuan pengambilan keputusan. Untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar, istilah Knowledge Discovery in Database (KDD) dan data mining sering digunakan [13].

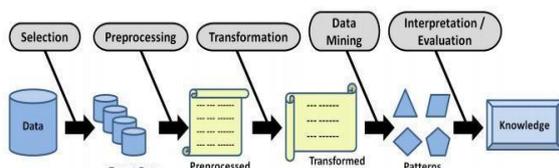
2.6. Support Vector Machine (SVM)

Vapnik, Guyon, dan Boser mengembangkan SVM. SVM pertama kali ditampilkan pada Annual Workshop on Computational Learning Theory sekitar tahun 1992 [14]. SVM saat ini merupakan pengklasifikasi yang banyak digunakan untuk berbagai tujuan klasifikasi. Kelebihan SVM termasuk kemampuan mereka untuk menyelesaikan masalah overfitting, menemukan solusi lokal yang ideal, memiliki rasio konvergensi rendah, dan memiliki kemampuan tinggi untuk generalisasi dalam kasus kecil [15].

2.7. Particle Swarm Optimization (PSO)

Salah satu metode komputasi evolusioner, Algoritma Particle Swarm Optimization (PSO), diperkenalkan pertama kali oleh Kennedy dan Eberhart pada tahun 1995. Perilaku sosial binatang adalah inspirasinya, seperti swarm burung. Algoritma melakukan penelusuran populasi dan dimulai dengan populasi random yang disebut partikel. Dengan memperbaharui generasi, solusi optimal dapat dicapai. Setelah melewati proses penelusuran, partikel mempunyai kecenderungan untuk bergerak ke area penelusuran yang lebih baik karena partikel tersebut bergerak melalui ruang dengan kecepatan dinamis yang disesuaikan dengan perilaku historisnya. Di dalam algoritma Particle Swarm Optimization (PSO), setiap partikel dihubungkan dengan kecepatan (velocity), yang membedakannya dari metode komputasi evolusioner lainnya [14].

3. METODE PENELITIAN



Gambar 1. Proses KDD

Metodologi yang digunakan pada penelitian ini ialah Knowledge Discovery in Database (KDD).

3.1. Selection

Pengumpulan data ulasan aplikasi Mobile Jaminan Kesehatan Nasional (JKN) yang dilakukan dengan Teknik scraping menggunakan library google play scraper menghasilkan sebanyak 3.000 data. Data yang telah terkumpul secara otomatis memiliki 11 atribut yaitu reviewId, username, userImage, content, score, thumbsUpCount, reviewCreatedVersion, at, replyContent, repliedAt dan appVersion. Selanjutnya dilakukan penyeleksian atribut yang digunakan yaitu "score" dan "content". Kemudian dilakukan pelabelan data secara manual.

3.2. Preprocessing

Data mentah yang dikumpulkan pada tahap preprocessing ini akan diubah menjadi data yang dapat digunakan pada tahap berikutnya. Preprocessing adalah proses awal yang akan mengubah data yang dimasukkan menjadi format yang tepat dan siap untuk diproses. Beberapa proses yang dilakukan selama preprocessing adalah cleansing, case folding, tokenization, dan filtering (stopword removal). Preprocessing tahap terdiri dari empat proses, termasuk:

- a. Cleansing adalah proses yang bertujuan untuk menghapus angka, symbol, tanda baca, emoticon dan menghapus single char.
- b. Case folding adalah proses yang bertujuan untuk membuat huruf kapital menjadi huruf kecil (lowercase) agar lebih terstruktur dan mudah diproses.
- c. Tokenization juga dikenal sebagai tokenisasi, adalah proses yang dirancang untuk memenuhi kebutuhan pemrosesan dengan memotong setiap kalimat ke dalam sekumpulan kata tunggal.
- d. Stopword bertujuan untuk pemilihan kata-kata yang tidak digunakan dalam proses identifikasi, seperti kata yang, kecuali, tetapi, dan, dapat, dan lain-lain [12].

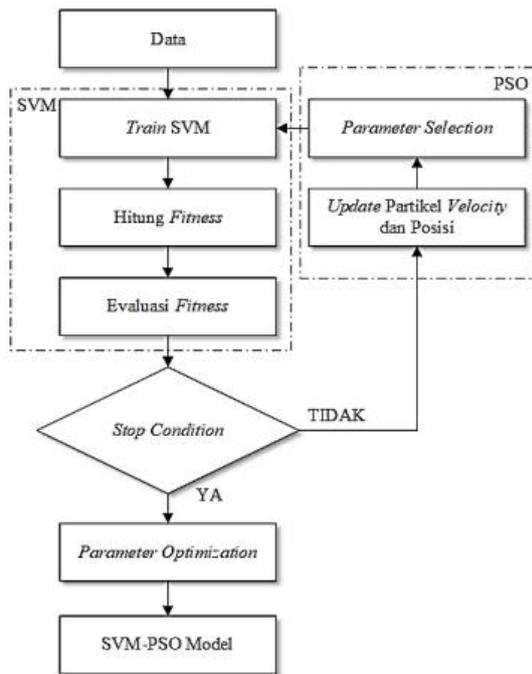
3.3. Transformation

Fase ini mengubah data opini menjadi data numerik dengan memberi bobot pada setiap kata atau istilah. Penelitian ini menggunakan vector TF-IDF yang terdapat pada operator Process Document From Data, diimplementasikan menggunakan tools RapidMiner.

3.4. Data Mining

Tujuan dari tahap ini adalah proses klasifikasi menggunakan metode SVM yang dioptimasi dengan PSO. Tujuannya adalah untuk menghasilkan nilai akurasi, presisi, recall, dan AUC dari metode klasifikasi SVM berbasis PSO. Gambar 2 mengilustrasikan model SVM berbasis PSO. Analisis SVM dimulai dengan mengubah teks menjadi data vektor. Karena sulit menentukan parameter yang

sesuai untuk SVM, artikel ini menggunakan PSO untuk mengatur parameter SVM yang dioptimalkan.



Gambar 2. Model SVM berbasis PSO

4. HASIL DAN PEMBAHASAN

Penelitian ini menggunakan Bahasa pemrograman *Python* pada *Jupyter Notebook* untuk pengambilan data ulasan di platform *Google Play Store*, kemudian dilanjutkan dengan tahap pengujian yang dilakukan pada *RapidMiner Educational 10.1*.

4.1. Selection

Pengumpulan data ulasan aplikasi Mobile Jaminan Kesehatan Nasional (JKN) yang dilakukan dengan teknik *scraping* menggunakan *library google play scraper* menghasilkan sebanyak 3.000 data. Secara otomatis dataset yang terkumpul memiliki 11 atribut yaitu *reviewId*, *username*, *userImage*, *content*, *score*, *thumbsUpCount*, *reviewCreatedVersion*, *at*, *replyContent*, *repliedAt* dan *appVersion*. Selanjutnya dilakukan penyeleksian atribut yang digunakan hanya *score* dan *content*, alasan penggunaan atribut tersebut yaitu atribut *score* akan digunakan pada proses pelabelan, sedangkan atribut *content* berisi teks *review* yang akan digunakan untuk analisis data.

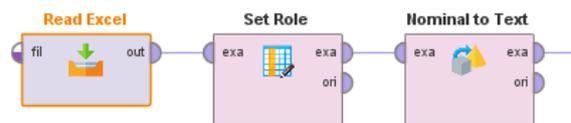
Kemudian dilakukan pelabelan data secara manual dan divalidasi oleh peneliti sendiri pada *Microsoft excel*. Data pada *excel* diurutkan berdasarkan *score* dari yang terkecil hingga terbesar dengan *score* terkecil yaitu 1 dan *score* terbesar yaitu 5. Pelabelan dilakukan menggunakan metode *rating based labelling* berdasarkan *score*, dimana *score* 1 sampai 3 didefinisikan sebagai *negative sentiment* serta *score* 4 dan 5 didefinisikan sebagai *positive sentiment*. Kelebihan metode ini adalah mudah diterapkan dan digunakan. Namun, nilai rating tidak selalu menunjukkan bahwa *review* dengan rating yang

baik akan memiliki sentimen positif, misalnya pada *review* yang mengandung frase *negative*. Pelabelan data tersebut menghasilkan 520 ulasan positif dan 2.480 ulasan *negative*. Langkah selanjutnya data dimasukkan ke dalam *rapidminer*, terlebih dahulu dilakukan penyeleksian atribut pada *Microsoft excel*. Atribut yang terdapat pada data untuk dimasukkan ke dalam *rapidminer* yaitu *label* dan *content*.



Gambar 3. Tampilan operator *read excel* pada *rapidminer*

Data yang sudah dilakukan pelabelan dibaca menggunakan operator *read excel*. Operator *read excel* ditampilkan pada gambar 3. Proses selanjutnya yaitu merubah format kolom. Pada format ini dilakukan perubahan tipe pada *attribute label* menjadi tipe *binominal* dan perubahan *role* menjadi *label*.



Gambar 4. Operator *set role* dan *nominal to text*

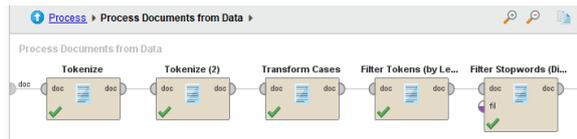
Setelah merubah format, ditambahkan operator *set role* dan *nominal to text*. Gambar 4 menampilkan operator *set role* dan *nominal to text*. Sedangkan operator *nominal to text* digunakan untuk mengubah jenis atribut nominal yang dipilih menjadi teks dan juga memetakan semua nilai atribut ke nilai *string* yang sesuai.

4.2. Preprocessing

Data yang diperoleh dalam penelitian biasanya belum siap digunakan dalam proses pengujian. Tahap preprocessing mengubah data yang diperoleh dari sumber data menjadi kumpulan data yang dapat diolah lebih lanjut dalam proses pengujian data. Tujuan dari prapemrosesan adalah untuk menghilangkan *noise* dari data dan melakukan beberapa langkah untuk mengubah data tidak terstruktur menjadi data terstruktur.

Pada tahap ini dilakukan proses *cleaning*, *case folding*, *tokenizing*, dan *filtering (stopword removal)* pada *RapidMiner*. *Cleaning* merupakan proses pembersihan teks yang dilakukan dengan menghilangkan *noise*. Pada tahap ini, *symbol*, angka, *emoticon* dan tanda baca yang tidak diperlukan untuk analisis sentimen akan dihilangkan. *Case folding* merupakan metode untuk mengubah seluruh kata pada teks menjadi kata-kata dalam huruf kecil. Dengan melakukan *case folding* ini, kata-kata yang sama akan melebur dan mengurangi dimensi permasalahannya [6]. *Tokenizing* merupakan metode yang digunakan

untuk memecah teks pada kalimat menjadi kata-kata atau frase yang membentuknya [6]. *Filter tokens by length* digunakan untuk memfilter token berdasarkan panjangnya, jumlah karakter yang dikandungnya antara 4 sampai 25 karakter. *Stop-word* merupakan tahap penghapusan kata yang tidak relevan terhadap penentuan klasifikasi sentimen jika termasuk di dalam daftar kata *stopword* [8]. Kamus *stopword* didapatkan dari *Kaggle*, berikut adalah linknya : *Indonesian Stoplist (kaggle.com)*. Gambar 5 menampilkan Subproses Operator *Process Document From Data*.



Gambar 5. Subproses operator *process document from data*

Subproses dari operator *process document from data* yaitu dilakukan *tokenize* untuk menghilangkan angka, menghilangkan karakter tanda baca, *transform cases* digunakan untuk merubah huruf kapital menjadi huruf kecil, *filter tokens by length* untuk menghapus kata yang kurang dari 4 huruf dan lebih dari 25 huruf, dan *filtering (stopword removal)* untuk menghapus kata-kata yang tidak memiliki makna penting. Tabel 1 menampilkan *output* dari *text preprocessing*.

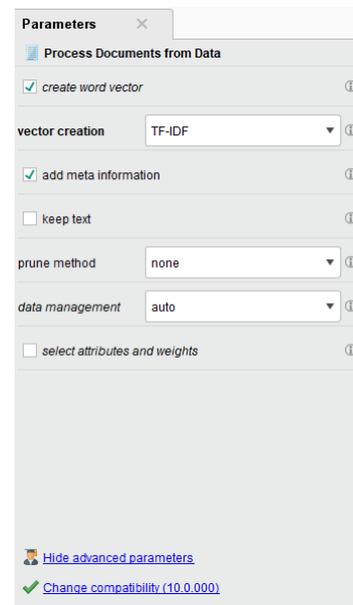
Table 1. *Output* dari *text preprocessing*

Proses	Label	Output
Sentimen asli	NEGATIF	Setelah diupdate kok jadi eror gini ya, uda 2 minggu dicoba, diuninstal berulang-berulang jg ttp aja eror. Berulang-ulang nyoba buka kartu peserta gak bisa, akhirnya aku uninstal berkali-kali tp pas mau login selalu muncul "Transaksi belum dapat diproses, Silahkan coba beberapa saat lagi" padahal jaringan aku sangat bagus. Mohon dong diperbaiki
	POSITIF	Bagus dan mudah dioperasikan 🤖 🤖 🤖 🤖
Cleaning dan case folding	NEGATIF	setelah diupdate kok jadi eror gini ya uda minggu dicoba diuninstal berulangberulang jg ttp aja eror berulangulang nyoba buka kartu peserta gak bisa akhirnya aku uninstal berkalkali tp pas mau login selalu muncul transaksi belum dapat diproses silahkan coba beberapa saat lagi padahal jaringan aku sangat bagus mohon dong diperbaiki
	POSITIF	bagus dan mudah dioperasikan

Proses	Label	Output
Tokenizing, filter tokens by length dan filter stopword	NEGATIF	'eror', 'gini', 'minggu', 'dicoba', 'diuninstal', 'berulangberulang', 'eror', 'berulangberulang', 'nyoba', 'buka', 'kartu', 'peserta', 'uninstal', 'berkalikali', 'login', 'muncul', 'transaksi', 'diproses', 'silahkan', 'coba', 'jaringan', 'bagus', 'mohon', 'diperbaiki'
	POSITIF	'bagus', 'mudah', 'dioperasikan'

4.3. Transformation

Pada tahap transformasi ini menggunakan operator *Process Document From Data*. Tahap ini merubah data teks menjadi numerik menggunakan *vector creation Term Frequency-Inverse Document Frequency (TFIDF)*. Gambar 7 menampilkan hasil *WordList (Process Document From Data)*.



Gambar 6. Parameter operator *process document from data*

Gambar 6 menampilkan parameter operator *process document from data* menggunakan *vector creation Term Frequency-Inverse Document Frequency (TFIDF)*. Pada parameter tersebut digunakan *create word vector*, dan *add meta information*, sedangkan untuk *prune method*nya adalah *none* dan *data management*nya adalah *auto*.

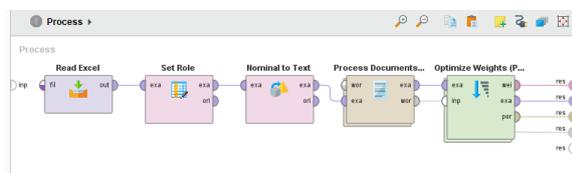
Word	Attribute Name	Total Occurrences ↓	Document Occurrences	NEGATIF	POSITIF
aplikasi	aplikasi	1688	1247	1486	202
daftar	daftar	863	679	814	49
login	login	691	532	653	38
update	update	584	462	528	56
masuk	masuk	558	455	520	38
tolong	tolong	479	444	442	37
data	data	355	292	280	75
kode	kode	338	286	319	19
susah	susah	324	291	301	23
nomor	nomor	321	221	291	30
salah	salah	300	270	284	16
faskes	faskes	294	239	231	63
muncul	muncul	289	241	270	19
bpjs	bpjs	284	241	214	70

Gambar 7. WordList (process document from data)

Pada gambar di atas menampilkan 14 kata yang memiliki frekuensi tertinggi yang terdiri dari kata aplikasi, daftar, login, update, masuk, tolong, data, kode, susah, nomor, salah, faskes, muncul, dan bpjs.

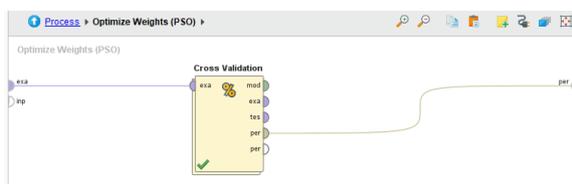
4.4. Data Mining

Tahap ini meliputi teknik *data mining* menggunakan algoritma klasifikasi pada *RapidMiner Studio Educational 10.1*. Hasil pengujian model yang dilakukan adalah klasifikasi *review* positif dan negatif menggunakan algoritma SVM dan pemilihan fitur PSO. Struktur model ini dalam penerapannya ditunjukkan pada Gambar 8.



Gambar 8. Desain model SVM berbasis PSO

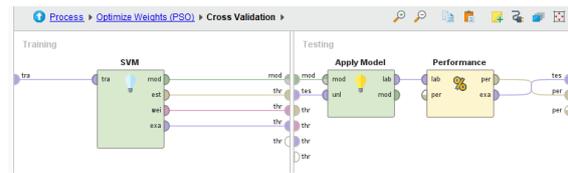
Gambar 8 menggambarkan rangkaian model pengujian secara global. Selanjutnya digunakan metode validasi silang untuk memvalidasi model terhadap algoritma SVM berbasis PSO. Pada tahap pengujian untuk klasifikasi berbasis PSO operator langsung ditambahkan di awal model seperti terlihat pada gambar 8. Gambar 9 menampilkan model pengujian validasi SVM berbasis PSO bagian 2.



Gambar 9. Model pengujian validasi SVM berbasis PSO bagian 2

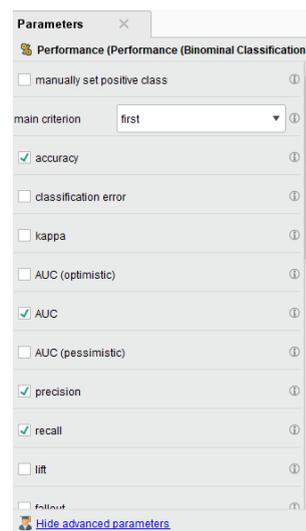
Di dalam operator PSO terdapat operator *cross validation* yang kemudian dilanjutkan penambahan operator SVM untuk memproses *training data* dan *testing data*. Di dalam operator *cross validation* terdapat algoritma yang digunakan yaitu SVM.

Gambar 10 menampilkan tahap percobaan validasi SVM berbasis PSO di dalam *cross validation*.



Gambar 10. Tahap percobaan validasi SVM berbasis PSO pada *cross validation*

Dalam subproses *cross validation* terdapat dua bagian yakni pelatihan dan pengujian, seperti yang ditunjukkan pada gambar 10. Tujuannya adalah untuk menggunakan algoritma SVM untuk membuat model dataset dengan membagi data menjadi 10:90, yang merupakan sepuluh persen data pengujian dan sembilan puluh persen data pelatihan. Data pengujian juga memiliki dua fitur, yang pertama adalah penerapan model, yang digunakan untuk menerapkan model yang telah dilatih sebelumnya dengan data pengujian.



Gambar 11. Parameter *performance*

Yang kedua ada fitur *performance* di bagian pengujian, yang digunakan untuk mengevaluasi hasil kinerja algoritma SVM yang menggunakan parameter pengukuran *confusion matrix* (akurasi, *recall*, dan presisi) serta AUC. Gambar 11 menampilkan parameter *performance (Binominal Classification)*.

4.5. Evaluation

Kemudian pada tahapan selanjutnya ialah proses evaluasi dari kinerja masing-masing algoritma yang sudah ditentukan menggunakan Teknik *confusion matrix* dan *ROC Curve*. Salah satu fungsi dari *confusion matrix* adalah untuk mengukur tingkat akurasi, presisi, dan *recall* dari model algoritma yang dievaluasi. Nilai akurasi adalah tingkat ketepatan presentase antara nilai prediksi dan nilai sebenarnya, nilai presisi adalah nilai akurasi dengan kelas yang

diprediksi, dan nilai *recall* adalah presentase nilai kinerja keberhasilan algoritma yang digunakan. Sedangkan, nilai AUC digunakan karena akan menunjukkan angka yang terkait langsung dengan data. Nilai AUC menunjukkan hasil pengukuran keseluruhan atas kesesuaian model yang digunakan. Nilai AUC yang lebih tinggi menunjukkan bahwa variabel yang diteliti memiliki kemampuan prediksi kejadian yang lebih baik [7].

Bagian ini menampilkan hasil *Confusion Matrix* pada akurasi, presisi dan *recall* dan AUC yang dihasilkan dari pengujian SVM berbasis PSO dengan memperhatikan parameter *population size* dan *bobot inersia* yang sudah dirancang sebelumnya. Pengujian ini menerapkan data *review* sebagai data uji untuk diterapkan pada algoritma SVM menggunakan metode optimasi PSO.

1. Accuracy

accuracy: 89.53% +/- 0.61% (micro average: 89.53%)

	true NEGATIF	true POSITIF	class precision
pred. NEGATIF	2447	281	89.70%
pred. POSITIF	33	239	87.87%
class recall	98.67%	45.96%	

Gambar 12. Hasil akurasi dari algoritma SVM berbasis PSO

Gambar 12 menunjukkan nilai akurasi yang diperoleh sebesar 89,53% dengan toleransi kesalahan sebesar 0,61%. Terdapat 239 *records* nilai *true* positif dan nilai *true negative* 2.447 *records*, yang menunjukkan bahwa data yang diklasifikasikan menurut sentimen positif sebanyak 239 *records* dan data yang diklasifikasikan menurut sentimen negatif sebanyak 2.447 *records*. Sedangkan data yang seharusnya sentimen positif namun diklasifikasikan *negative* (*false negative*) sebanyak 33 *records* dan data yang seharusnya diklasifikasikan sentimen *negative* namun diklasifikasikan positif (*false positive*) sebanyak 281 *records*.

2. Precision

precision: 88.17% +/- 4.61% (micro average: 87.87%) (positive class: POSITIF)

	true NEGATIF	true POSITIF	class precision
pred. NEGATIF	2447	281	89.70%
pred. POSITIF	33	239	87.87%
class recall	98.67%	45.96%	

Gambar 13. Hasil *precision* dari algoritma SVM berbasis PSO

Nilai presisi yang diperoleh sebesar 88,17% dengan toleransi kesalahan sebesar 4,61% ditunjukkan pada gambar 13.

3. Recall

recall: 45.96% +/- 3.56% (micro average: 45.96%) (positive class: POSITIF)

	true NEGATIF	true POSITIF	class precision
pred. NEGATIF	2447	281	89.70%
pred. POSITIF	33	239	87.87%
class recall	98.67%	45.96%	

Gambar 14. Hasil *recall* dari algoritma SVM berbasis PSO.

Nilai *recall* yang diperoleh sebesar 45,96% dengan toleransi kesalahan sebesar 3,56% ditunjukkan pada gambar 14.

4. Nilai AUC



Gambar 15. Hasil kurva ROC

Kurva *Receiver Operating Characteristic* (ROC), yang ditunjukkan pada Gambar 15, dihasilkan dari hasil pengujian kinerja, dengan nilai *Area Under Curve* (AUC) sebesar 0,869 yang merupakan klasifikasi yang baik.

attribute	weight ↓
perbaikannya	1.000
medika	1.000
laksanakan	1.000
nomenya	0.999
awam	0.999
polos	0.998
woooiii	0.998
kaji	0.998
kabanyakan	0.998
melayani	0.998
sekalinnya	0.998
pengerjaan	0.998
team	0.997
ngapee	0.997

Gambar 16. *Attribute weights* (optimize weights (PSO))

Pada gambar 16 di atas, nilai weightnya berada diantara 1 dan 0 sesuai dengan parameter yang telah ditentukan sebelumnya. Nilai 0 merupakan fitur yang tidak terpilih.

5. KESIMPULAN DAN SARAN

Pengujian terhadap data hasil *scraping* dari *google play store* telah dilakukan. Berdasarkan hasil analisis sentimen terhadap aplikasi *Mobile JKN* di *google play store* dengan menggunakan algoritma SVM yang dioptimalisasi menggunakan PSO, dapat disimpulkan bahwa berdasarkan pelabelan kelas sentimen, jumlah ulasan positif sebanyak 520 dan

ulasan *negative* sebanyak 2.480. Hasil menunjukkan bahwa algoritma SVM dengan optimasi PSO memiliki nilai akurasi, presisi, *recall*, dan AUC masing-masing sebesar 89,53%, 88,17%, 45,96%, dan 0,869. Terbukti bahwa PSO dapat meningkatkan kinerja SVM dengan kemampuan pemilihan fitur. Berdasarkan temuan penelitian, penulis membuat rekomendasi untuk digunakan dalam penelitian selanjutnya, yaitu: penelitian lebih lanjut harus dilakukan untuk kasus yang sama dengan menggunakan operator *feature selection by corellation* atau dengan *feature selection by gain information*, sehingga kinerja keduanya dapat dibandingkan dan diperoleh nilai akurasi terbaik.

DAFTAR PUSTAKA

- [1] M. Nurmalasari, N. A. Temesvari, and ..., "Analisis Sentimen terhadap Opini Masyarakat dalam Penggunaan Mobile-JKN untuk Pelayanan BPJS Kesehatan Tahun 2019," *Indones. Heal. ...*, 2020, [Online]. Available: <https://inohim.esaunggul.ac.id/index.php/INO/article/view/208>
- [2] N. Bethry, B. Tjikdaphia, and Sulastri, "Comparison of nbc, svm, knn classification results in sentiment analysis of mobile jkn," *JURTEKSI (Jurnal Teknol. dan Sist. Informasi)*, vol. IX, no. 4, 2023, doi: <https://doi.org/10.33330/jurteksi.v9i4.2539>.
- [3] A. Filemon, H. Kaban, and N. Yudistira, "Analisis Sentimen Aplikasi E-Government berdasarkan Ulasan Pengguna menggunakan Metode Maximum Entropy dan Seleksi Fitur Mutual Information," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 4, pp. 1452–1458, 2021.
- [4] C. Annisa, M. Afdal, and T. K. Ahsyar, "Perbandingan Algoritma Naive Bayes Classifier Dan K-Nearest Neighbor Pada Sentimen Review Aplikasi Mobile Jkn," *J. Media Inform. Budidarma*, vol. 7, pp. 1033–1040, 2023, doi: [10.30865/mib.v7i3.6242](https://doi.org/10.30865/mib.v7i3.6242).
- [5] P. Arsi, R. Wahyudi, and R. Waluyo, "Optimasi SVM Berbasis PSO pada Analisis Sentimen Wacana Pindah Ibu Kota Indonesia," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 1, no. 10, pp. 231–237, 2021.
- [6] N. Hafidz and D. Y. Liliana, "Klasifikasi Sentimen pada Twitter Terhadap WHO Terkait Covid-19 Menggunakan SVM, N-Gram, PSO," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 1, no. 10, pp. 3–4, 2021.
- [7] A. Faisal, Y. Alkhalifi, A. Rifai, and W. Gata, "Analisis Sentimen Dewan Perwakilan Rakyat Dengan Algoritma Klasifikasi Berbasis Particle Swarm Optimization," *JOINTECS (Journal Inf. Technol. Comput. Sci.)*, vol. 5, no. 2, p. 61, 2020, doi: [10.31328/jointecs.v5i2.1362](https://doi.org/10.31328/jointecs.v5i2.1362).
- [8] E. R. Indriyani and M. Wibowo, "Perbandingan Metode Naive Bayes dan Support Vector Machine Untuk Analisis Sentimen Terhadap Vaksin Astrazeneca di Twitter," *J. MEDIA Inform. BUDIDARMA*, vol. 6, no. September 2020, pp. 1545–1553, 2022, doi: [10.30865/mib.v6i3.4220](https://doi.org/10.30865/mib.v6i3.4220).
- [9] S. N. Hakim, A. J. Putra, and A. U. Khasanah, "Sentiment analysis on myindihome user reviews using support vector machine and naive bayes classifier method," *Int. J. Ind. Optim.*, vol. 2, no. 2, p. 141, 2021, doi: [10.12928/ijio.v2i2.4449](https://doi.org/10.12928/ijio.v2i2.4449).
- [10] D. A. Kristiyanti and S. Hardani, "Sentiment Analysis of Public Acceptance of Covid-19 Vaccines Types in Indonesia using Naive Bayes, Support Vector Machine, and Long Short_Term Memory (LSTM)," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 158, pp. 2–6, 2023.
- [11] V. W. D. Thomas and F. Rumaisa, "Analisis Sentimen Ulasan Hotel Bahasa Indonesia Menggunakan Support Vector Machine dan TF-IDF," *J. Media Inform. Budidarma*, vol. 6, no. 3, p. 1767, 2022, doi: [10.30865/mib.v6i3.4218](https://doi.org/10.30865/mib.v6i3.4218).
- [12] M. R. Fahlevvi, "Analisis Sentimen Terhadap Ulasan Aplikasi Pejabat Pengelola Informasi Dan Dokumentasi Kementerian Dalam Negeri Republik Indonesia Di Google Playstore Menggunakan Metode Support Vector Machine," *J. Teknol. dan Komun. Pemerintah.*, vol. 4, no. 1, pp. 1–13, 2022, doi: [10.33701/jtkp.v4i1.2701](https://doi.org/10.33701/jtkp.v4i1.2701).
- [13] I. W. M. Putri, R. Rusdah, L. Suryadi, and D. Anubhakti, "Prediksi Kelulusan Mahasiswa Fakultas Teknologi Informasi ISB Atma Luhur Menggunakan Algoritma C4.5," *J. SISFOKOM (Sistem Inf. dan Komputer)*, vol. 12, no. 03, pp. 363–369, 2023, doi: [10.32736/sisfokom.v12i3.1731](https://doi.org/10.32736/sisfokom.v12i3.1731).
- [14] E. Nurlelah and Y. D. Utami, "Seleksi Atribut pada Algoritma Neural Network Menggunakan Particle Swarm Optimization Untuk Diagnosis Penyakit Liver," *Kumpul. J. Ilmu Komput.*, vol. 09, no. 02, pp. 353–365, 2022.
- [15] V. K. S. Que, A. Iriani, and H. D. Purnomo, "Analisis Sentimen Transportasi Online Menggunakan Support Vector Machine Berbasis Particle Swarm Optimization," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 9, no. 2, pp. 162–170, 2020, doi: [10.22146/jnteti.v9i2.102](https://doi.org/10.22146/jnteti.v9i2.102).